



Canasta

basket **canasta** canestro corbeille Korb кошик 筐子 سلة टोकरी 籠 корзина

# State of Canasta and Canasta for SMW

Jeffrey Wang

SMWCon Fall 2022 – 27 October 2022 – Breda, Netherlands

# Pain points of vanilla MediaWiki



Canasta

- Most common way is to run on a LAMP stack
  - Old and outdated
- Not easily installable to maximize experience
  - Hard to install: Elasticsearch, VisualEditor (formerly), etc.
- Not easily scalable
- No built-in backup solution
- Doesn't come with many extensions or skins
- Not easily upgradable
- No automated cron jobs to run important maintenance scripts
- Configuration drift easily occurs



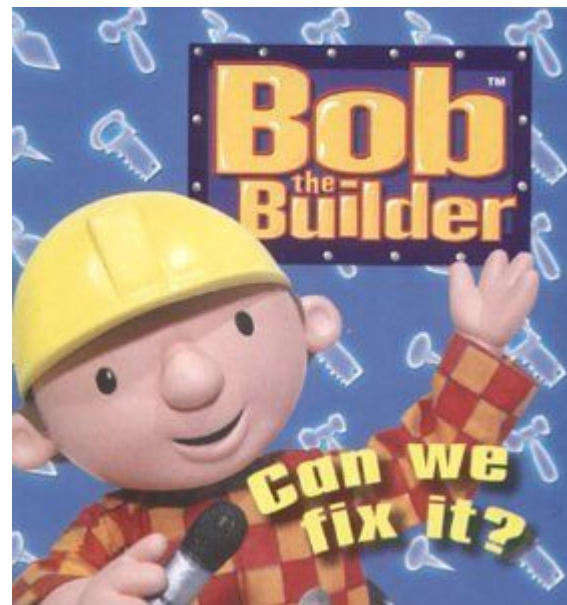
# Can we fix it?



Canasta

A solution that:

- Is easy to install, upgrade, and maintain
- Is scalable
- Contains built-in support for automation and backups
- Eliminates configuration drift



# Can we fix it? **Yes, we can!**



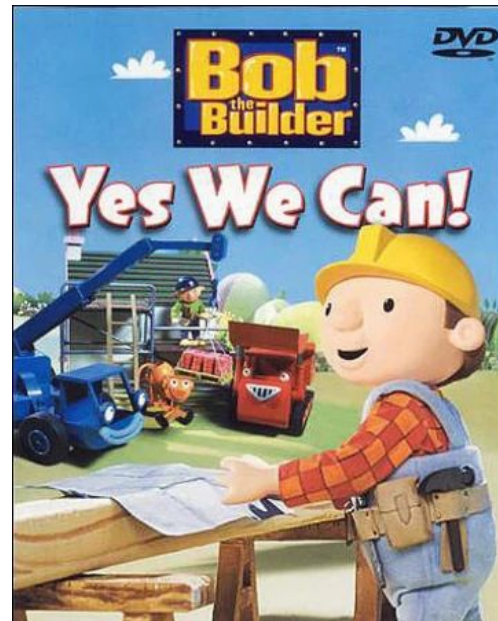
Canasta

A solution that:

- Is easy to install, upgrade, and maintain
- Is scalable
- Contains built-in support for automation and backups
- Eliminates configuration drift

Turns out, this is **all possible**.

It's called **Canasta**.



# What's Canasta?



Canasta

- “A comfy home for your data”
- All-in-one MediaWiki package
  - Docker-based stack powering MediaWiki
  - 100+ extensions already included
  - Data structures and packages ready for install
- Reducing technical and administrative complexity
  - Spend less time setting up, gain back more time for using your wiki



# A Docker solution for MediaWiki



Canasta

- Before Canasta, no open-source, production-ready Docker image for running MediaWiki existed
  - Community had to create own Docker image or use WMF image (despite it only officially being made for testing)
- Today, Canasta is the go-to Docker distribution for the Enterprise MediaWiki community
  - Support for Docker Compose and Kubernetes
  - Support latest MediaWiki LTS versions
  - >100 extensions and skins
  - Easy to install

Canasta tech stack

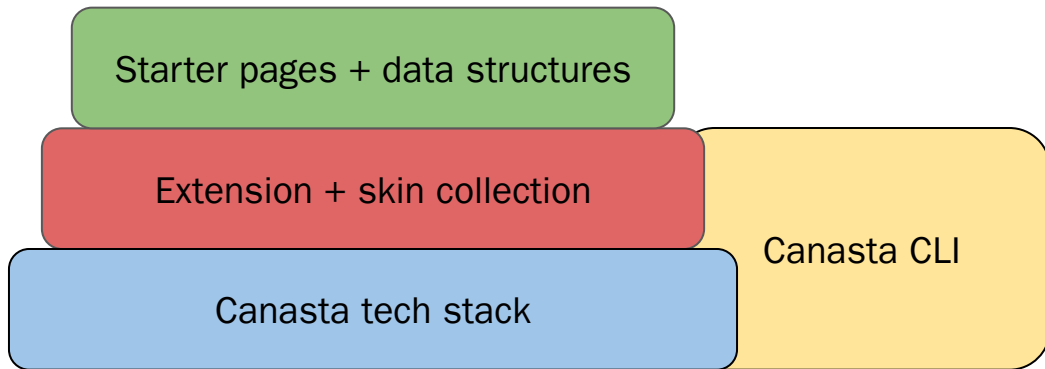
Extension + skin collection

Starter pages + data structures

# Components of Canasta



Canasta



# Canasta stack's values



Canasta

- Fully open-source
- Convenient
- Same core MediaWiki
- Stable
- Customizable
- Extensible
- Source control compatible

Canasta tech stack

Extension + skin collection

Starter pages + data structures

# Loading bundled extensions and skins



Canasta

It's as easy as 1-2-3:

1. Pick the extension you want to load from the list at [canasta.wiki](https://canasta.wiki).
2. Using the Canasta CLI, run `canasta extension enable`
  - a. Or in `LocalSettings.php` or a new file in `settings/` folder, add the following (with config settings too):
    - i. For extensions: `wfLoadExtension( 'PageForms' );`
    - ii. For skins: `wfLoadSkin( 'Timeless' );`
3. Restart the Canasta container.

That's it!

Canasta tech stack

Extension + skin collection

Starter pages + data structures

# Advantages of the Canasta stack



Canasta

- Easier, faster to set up
- Low maintenance requirements
  - Easy to upgrade MediaWiki
- Eliminate configuration drift
- Scalable and performant
- Written as a compatibility layer over pure MediaWiki
  - Future versions of MediaWiki should be easy for Canasta to support
- Easy to migrate back and forth between Canasta and non-Canasta
- Reduced development time and overhead
  - Canasta stakeholders all can contribute to each others' success

Canasta tech stack

Extension + skin collection

Starter pages + data structures

# Switching to Canasta



Canasta

- Super easy!
- Drag and drop your existing LocalSettings.php
  - Except changing your database if you want to use the built-in Canasta DB
- Ditch the complicated setup of CirrusSearch, VisualEditor, Varnish, and HTTPS

Canasta tech stack

Extension + skin collection

Starter pages + data structures

# Upgrading MediaWiki

## Vanilla:

1. Backup database and existing files
2. Download new files
3. Replace files
4. `chown?? chmod?? !!!`
5. Download new extension files
6. Replace extension files
7. Run `update.php` or use web upgrader

Canasta tech stack

Extension + skin collection

Starter pages + data structures



# Canasta

## Canasta (with Docker Compose):

1. Pull latest stack repo version (`git pull`)
  2. Change image tag on Docker Compose YAML
  3. `docker compose pull && docker compose down && docker compose up -d`
- **Coming soon: CLI command**
    - As simple as `canasta upgrade`

# Under the hood



Canasta

- Runs on Docker, orchestrated by either Docker Compose or Kubernetes
- Easy setup instructions, set everything up in under 15 minutes
- Stack comprised of:
  - Canasta application container
    - MediaWiki, Apache, PHP, job queue runner, maintenance scripts runner, media transcoder, automated DB backups, 100+ bundled extensions
  - MySQL container
  - Elasticsearch container (for CirrusSearch)
  - Caddy container (for HTTPS/TLS termination)
  - Varnish (for frontend caching)

Canasta tech stack

Extension + skin collection

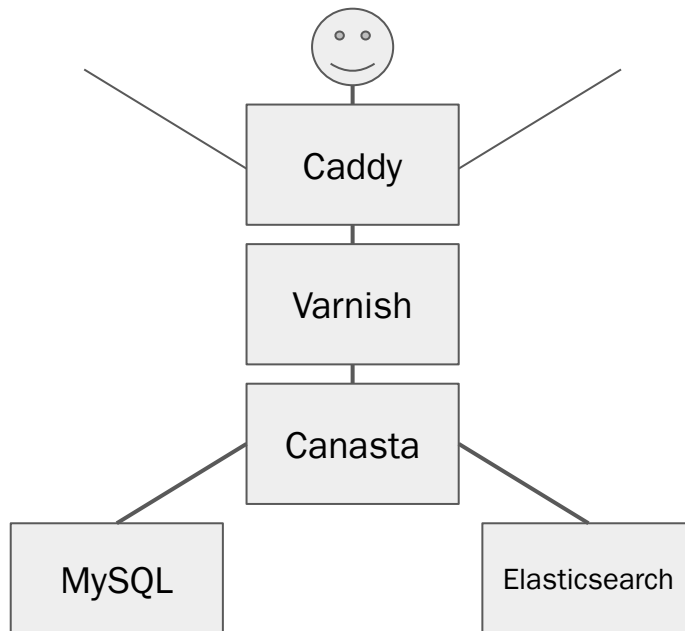
Starter pages + data structures

# The Canasta container stack



Canasta

(As of v1.0.0)



Canasta tech stack

Extension + skin collection

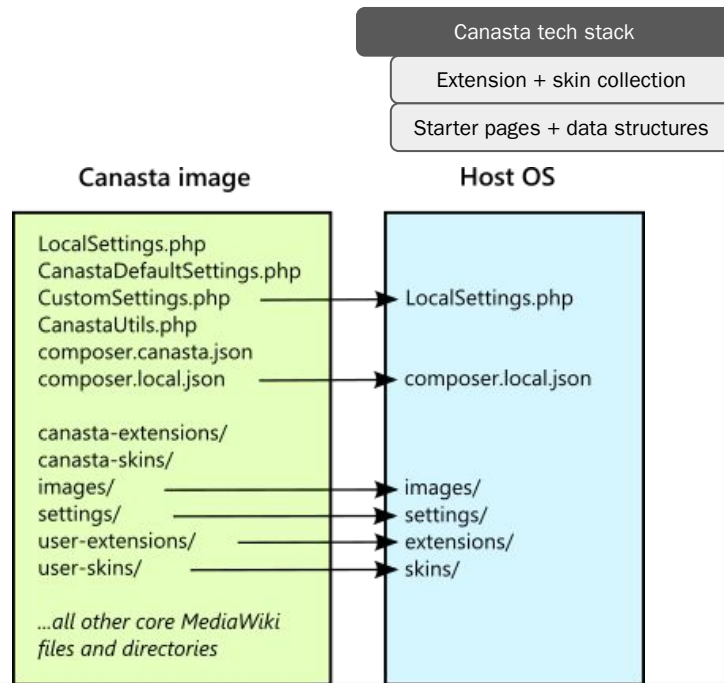
Starter pages + data structures

# More than meets the eye



Canasta

- Host OS shows all user-customizable MediaWiki files and directories
  - Same nomenclature as vanilla MediaWiki
- Canasta-specific code and shims + bundled extensions/skins located in Canasta image/container only
  - Not accessible to sysadmin—this prevents accidental changes contrary to how Canasta is meant to be used
  - Clear divide between persistent data and ephemeral code

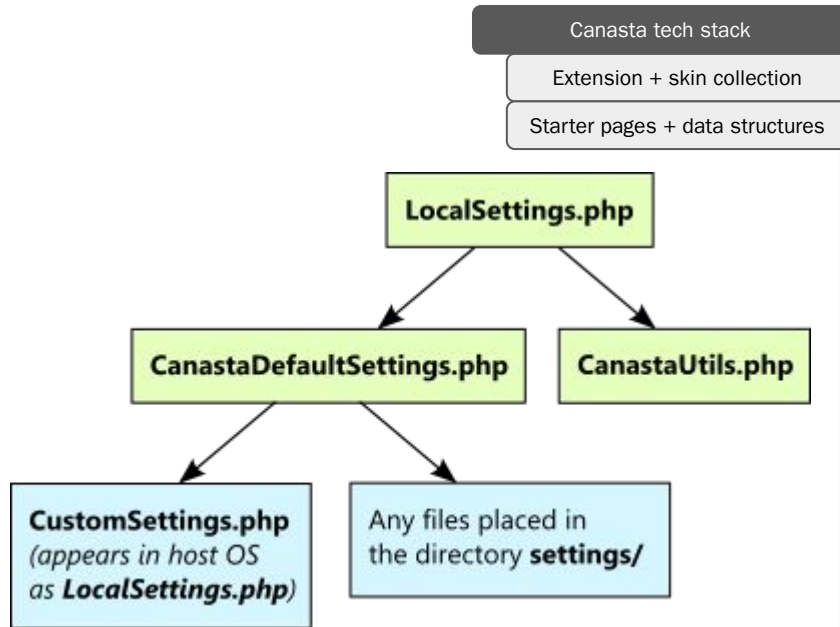


# LocalSettings “Inception”



Canasta

- What the sysadmin sees:
  - LocalSettings.php
  - Settings snippets under `settings/` directory
- What Canasta loads to LocalSettings.php
  - CanastaUtils.php (shims)
  - CanastaDefaultSettings.php (the absolute necessities for Canasta to work; the behind-the-scenes “plumbing”)



# Derivative images



Canasta

- Adapt Canasta to your needs
  - Start from the base Canasta image and make changes using a Dockerfile
- Making modifications is officially supported
- Things you could change
  - Bundled extensions/skins
  - Supporting services
  - CanastaDefaultSettings.php – change how the Canasta image does things
- Why make derivative images?
  - Maybe you manage many wikis but add your own special way of doing things.
  - Maybe you want to only use part of Canasta.

Canasta tech stack

Extension + skin collection

Starter pages + data structures

# Who's using the Canasta stack?



Canasta

Several companies in the MediaWiki industry have adopted the Canasta stack:

- WikiWorks
- WikiTeq
- MyWikis
- ...and more? :)

**WikiWorks**   
MediaWiki consulting

 **WikiTeq** 

 **mywikis**

Canasta tech stack

Extension + skin collection

Starter pages + data structures

# 100+ extensions and skins



# Canasta

- Curated and vetted
- Criteria for adding extensions and skins
  - Works with the version of MediaWiki used by Canasta (currently 1.35)
  - Stable
  - Actively maintained and low risk of abandonment
  - Provide unique value that isn't provided by any other bundled extension

## Skins included in Canasta

- Chameleon
- Cologne Blue
- Minima Neue
- Modern
- MonoBook
- Pivot
- Refreshed
- Timeless
- Vector

## Extensions included in Canasta

### Administrative tools

[Admin Links](#) · [Interwiki](#)

### Analytics

[ContributionScores](#) · [Editcount](#) · [GoogleAnalyticsMetrics](#) · [GTag](#) · [HotTentLinks](#) · [WholeSiteWatching](#)

### Automated page creation & modification

[Data Transfer](#) · [Page Exchange](#) · [Replace Text](#)

### Communication

[CommentsStreams](#) · [ContactPage](#) · [DiscussionTools](#) · [DismissableSiteNotice](#) · [Echo](#) · [GlobalNotice](#) · [MassMessage](#) · [MassMessageEmail](#) · [NewUserMessage](#) · [Thanks](#) · [WikiForum](#)

### Data storage and querying

[Cargo](#) · [Disambiguator](#) · [PageImages](#) · [Semantic Brackcung Links](#) · [Semantic Compound Queries](#) · [Semantic Drilldown](#) · [Semantic Extra Special Properties](#) · [Semantic MediaWiki](#) · [Semantic Result Formats](#) · [Semantic Scribunto](#) · [SimpleChanges](#) · [SubPageList](#)

### Editing

[CharInsert](#) · [CodeEditor](#) · [CodeMirror](#) · [InputBox](#) · [LinkSupport](#) · [Linter](#) · [Page Forms](#) · [SaveSpinner](#) · [TemplateData](#) · [TinyMCE](#) · [VEFormat](#) · [VisualEditor](#) · [WikiEditor](#)

### File upload & handling

[CommonsMetadata](#) · [GoogleDocCreator](#) · [ImageMap](#) · [MultiUpload](#) · [MultimediaViewer](#) · [PdfHandler](#) · [SimpleBatchUpload](#) · [TimedMediaHandler](#) · [UploadWizard](#)

### HTML & multimedia support

[EmbedVideo](#) · [Flex Diagrams](#) · [Graph](#) · [HTML Tags](#) · [Maze](#) · [Math](#) · [SimpleMathJax](#) · [Wikisets](#)

### Interface & display customization

[BetaFeatures](#) · [Gadgets](#) · [Header Footer](#) · [HeadScript](#) · [SkinPerPage](#)

### Meta tags

[Description2](#) · [OpenGraphMeta](#) · [WikiSEO](#)

### PDF export

[Collection](#)

### Read & write access

[Approved Revs](#) · [LockAuthor](#) · [Lockdown](#)

### Search

[AdvancedSearch](#) · [CirrusSearch](#) · [Elastic](#)

### Social tools

[AJAXPoll](#) · [SocialProfile](#) · [VoteNY](#)

### Spam control

[AbuseFilter](#) · [ConfirmEdit](#) · [Nuke](#) · [SmileScam](#) · [SpamBlacklist](#) · [TitleBlacklist](#)

Canasta tech stack

Extension + skin collection

Starter pages + data structures

### Translation

[LocalisationUpdate](#) · [UniversalLanguageSelector](#)

### User management

[CheckUser](#) · [LookupUser](#) · [Renameuser](#) · [UserMerge](#)

### User registration

[AntiSpool](#) · [ConfirmAccount](#) · [Email Authorization](#) · [LDAPAuthentication2](#) · [LDAP Authorization](#) · [LDAPProvider](#) · [OAuthAuth](#) · [OpenID Connect](#) · [PluggableAuth](#)

### Other wikitext additions

[Analyze](#) · [Cite](#) · [CiteThisPage](#) · [External Data](#) · [Header Tabs](#) · [LinkTarget](#) · [Lingo](#) · [Loops](#) · [MagicNoCache](#) · [MinifyDocs](#) · [MobileDetect](#) · [MyVariables](#) · [NameAlpha](#) · [ParameterFunctions](#) · [Poem](#) · [RegularToolbox](#) · [Scribunto](#) · [SyntaxHighlight](#) · [Variables](#) · [UriGetParameters](#)

### Non-wikitext display utilities

[Bootstrap](#) · [BootstrapComponents](#) · [Breadcrumb2](#) · [CategoryTree](#) · [MobileFrontend](#) · [Popups](#) · [SandboxLink](#) · [SecureLinkFixer](#)

### Other

[EventLogging](#) · [EventStreamConfig](#) · [Favorites](#) · [TextExtracts](#) · [RevisionSlider](#)

# Predefined data structures



Canasta

- Everything is made for you
- Page Exchange + Canasta = a match made in heaven
  - Included with Canasta
  - One line to enable, one more line for each package

```
cfLoadExtension( 'PageExchange' );  
$wgPageExchangePackageFiles[] =  
'https://raw.githubusercontent.com/CanastaWiki/CanastaPXConsultingCRM/main/page-exchange.json';  
$wgPageExchangePackageFiles[] = 'https://raw.githubusercontent.com/CanastaWiki/CanastaPXWidgets/main/page-exchange.json';
```

- Import, update, delete
  - No risk in trying a new package

Canasta tech stack

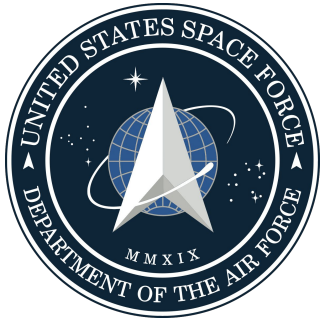
Extension + skin collection

Starter pages + data structures

# Who's considering/already using Canasta?



Canasta



# Canasta for Semantic MediaWiki



Canasta

- Semantic MediaWiki and other semantic extensions bundled in Canasta
  - Semantic Breadcrumb Links
  - Semantic Compound Queries
  - Semantic Drilldown
  - Semantic Extra Special Properties
  - Semantic Result Formats
  - Semantic Scribunto
- Easy to enable:

```
wfLoadExtension( 'SemanticMediaWiki' ); // for v4 and above  
  
enableSemantics( 'semantic.mywikis.eu' );
```

# Canasta makes SMW easy



Canasta

- Common problems
  - Upgrade key and refreshes
  - Running job queues
- Canasta solves by built-in automation
- Still possible to run maintenance scripts manually

```
docker compose exec web php /var/www/mediawiki/w/extensions/SemanticMediaWiki/maintenance/rebuildData.php
```

# Canasta + Open CSP



Canasta

Possible to install Open CSP on Canasta

Stay tuned for tomorrow's Open CSP presentations

# Updates since EMWCon in April 2022



Canasta

- A new Canasta CLI
- Kubernetes support (beta)
  - CLI does not support K8s yet
- Removal of cfLoadExtension/cfLoadSkin
  - Use wfLoadExtension/wfLoadSkin for both Canasta-bundled and user-installed extensions
- Support for latest minor releases of MediaWiki 1.35
- U.S. Department of Defense released “hardened” version of Canasta

# Canasta CLI



Canasta

- Easiest way to install, run, and manage Canasta wikis
- CLI = Command line interface
- Features
  - Create/delete wikis
  - Start/stop Canasta
  - Enable/disable extensions and skins
  - Backup and restore with Restic
  - Run maintenance scripts

# Using Canasta CLI



Canasta

- Each Canasta instance has an ID, unique to the machine where it's running
- Install with one command

```
sudo canasta create -i canastaId -n example.com -w Canasta Wiki -a admin -o docker-compose
```

- Or import an existing wiki with one command

```
sudo canasta import -i importWikiId -d ./backup.sql.gz -e ../.env -l ../LocalSettings.php
```

- Enable/disable extension

```
sudo canasta extension enable VisualEditor,PluggableAuth -i canastaId
```

More examples available at [canasta.wiki/cli](https://canasta.wiki/cli)

# Installing the CLI



Canasta

- Prerequisite: Docker and Docker Compose
- Run install command

Instructions available at [canasta.wiki/cli](https://canasta.wiki/cli)

# Technical information about Canasta CLI



Canasta

- Developed by Google Summer of Code contributor Amal Paul
- Written in Go
- Install Docker, then run one command to install

```
curl -fsL https://raw.githubusercontent.com/CanastaWiki/Canasta-CLI/main/install.sh | bash
```

- Use with sudo if Docker needs sudo to run
- Configuration stored in `/etc/canasta/conf.json`
- Works with Docker Compose; Kubernetes support coming soon
- We provide Linux x86 binary by default; others are self-serve

# Backing up your wiki



Canasta

## Vanilla:

- Many different ways to do it
- Need to set up manually
- Need to write scripts by yourself
- Headache to initialize

## Canasta:

- Backups capability built into CLI
- Uses Restic
- Backup to other server or cloud storage
- `canasta restic init`
- `canasta restic take-snapshot`
- `canasta restic restore`

# CLI demo



Canasta

Now, we'll set up a Canasta wiki from scratch using the CLI!

# Current development goals



Canasta

- MediaWiki 1.39 support
  - YAML configuration support
    - Should be facilitated by the removal of cfLoadExtension/cfLoadSkin
  - Updating bundled extensions and skins to include new defaults such as Minerva Neue
- Wiki farm support
  - Multiple wikis in the same Canasta container
- Multi-domain support via Traefik
  - Expected to replace Caddy

# What's next? (Canasta tech stack)



Canasta

- ARM (AArch64) support
- Wiki farm support
- Add Kubernetes support to Canasta CLI
- Improved support for cloud container services
  - AWS ECS
  - Azure Container Instances
- Upgrade stacks to use latest versions of everything for MW 1.39
  - PHP 8.0
  - Elasticsearch 7
  - and more...

Canasta tech stack

Extension + skin collection

Starter pages + data structures

# What's next? (Extensions + skins collection)



Canasta

- Add more extensions and skins
  - As long as they fit the criteria
- Criteria for adding extensions and skins
  - Works with the versions of MediaWiki used by Canasta (currently 1.35, soon to be 1.39)
  - Stable
  - Actively maintained and low risk of abandonment
  - Provide unique value that isn't provided by any other bundled extension
- Committee reviews extensions and skins for inclusion/exclusion
- Upgrade extensions to be compatible with MediaWiki 1.39
  - Identify and deprecate extensions and skins incompatible with 1.39

Canasta tech stack

Extension + skin collection

Starter pages + data structures

# What's next? (Starter pages + data structures)



Canasta

- Add more starter pages and data structures via Page Exchange
- Support for adding starter pages and data structures from the future Canasta CLI

Canasta tech stack

Extension + skin collection

Starter pages + data structures

# We need contributions!



Canasta

- Canasta couldn't be possible without open source community support
- What Canasta needs
  - **Developers.** Canasta was a large and ambitious project to start with. Now that it's stable, it still needs your help to add new features and expand its reach.
  - **Your input.** Feedback and suggestions from the community are crucial in deciding the direction of the project. Join the Canasta mailing list today!
    - **[canastawiki.googlegroups.com](https://canastawiki.googlegroups.com)**
  - **Funding.** To promote further development of Canasta, we need financial support to sustain our development initiatives.

# Project governance



Canasta

- Current state

- Yaron Koren, project founder, serves as “benevolent dictator”
- WikiWorks, WikiTeq, and MyWikis contribute development resources and set the direction
- Biweekly community cadence meetings intended for feedback from community members, stakeholders, and decision makers

- Future plans

- Who gets to decide the direction of development?
  - Every individual will have a say
  - Organizations giving financial support will also have a say
- Individuals and organizations are welcome to contribute in-kind through their development
- Concrete plans will be decided in the future

# More info



Canasta

- GitHub: [github.com/CanastaWiki/Canasta](https://github.com/CanastaWiki/Canasta)
  - File bug reports and feature requests on our GitHub Issues page
- Website: [canasta.wiki](https://canasta.wiki)
  - Official resource for learning about Canasta and how to use it

# Q&A session



Canasta

Now, we'll hold a Q&A session for the remaining time.

Ask about anything related to Canasta!